

Аплéт (англ. *Applet*) — коротка комп'ютерна програма, що функціонально розширює можливості основної програми, або інтернет-застосування. Наприклад, додає в веб-сторінку функцію анімації. Для створення аплетів в мережі Інтернет використовується мова програмування Java.

Java-аплет забезпечує можливість використовувати в World Wide Web «виконуваний вміст» шляхом виконання в гіпертекстовому документі міні-програм, відомих як **аплети**. Незвичайним є формат готового до виконання коду, у який транслюються аплети Java, — це апаратно-незалежний побайтовий формат. Завдяки використанню такого нейтрального формату відкомпільовані Java-програми можна переміщати по Internet, оскільки вони не залежать від операційної системи, у якій виконуються.

Переваги аплетів

Переваги використання Java-аплетів

- кроссплатформенність -аплет може працювати на «всіх» встановлених до цього часу версіях Java, а не тільки з останньою версією, а проте, якщо аплет вимагає останню версію JRE, то клієнт буде змушений чекати більш тривалої завантаження;
- аплет підтримується більшістю браузерів;
- він кешується в більшості браузерів, а тому буде швидко завантажуватися при поверненні на веб-сторінку; але аплет може зберігатися в кеші і створювати проблеми після виходу нових версій;
- він може мати повний доступ до машини, на якій виконується, якщо користувач згодний на це;
- аплет може поліпшити використання: після першого запуску аплету, коли JVM вже виконується і швидко запускається, переважно у постійних користувачів Java, проте JVM доведеться перезапускати кожен раз, коли запускається новий браузер;
- він може запуститися з порівнянною (але зазвичай повільніше) швидкістю на інших компільованих мовах, таких як C ++, але у багато разів швидше, ніж JavaScript;
- він може перенести роботу з сервера до клієнта, роблячи Інтернет-рішення з, великим числом користувачів / клієнтів.

Java-аплет володіє наступними недоліками:

- він вимагає установки Java-розширення (plug-in), яке не у всіх браузерах доступно за замовчуванням;
- він не може запуснитися до тих пір, поки не запусниться віртуальна Java-машина, і це може зайняти значний час при першому запуску;
- створення і дизайн хорошого користувальницького інтерфейсу з використанням аплетів вважається більш складним завданням, ніж за допомогою технології, заснованої на HTML;
- деякі організації допускають лише програмне забезпечення, встановлене адміністраторами. У результаті багато користувачів не можуть бачити аплети за замовчуванням.
- аплети можуть зажадати використання певного JRE.

Інтеграція аплету у веб-додаток

Примітка: Шаблонів проекту, спеціально розроблених для створення аплетів в середовищі IDE, не існує, проте їх можна розробляти в звичайному проекті Java і упаковувати в веб-проект відповідно до вказівок в цьому керівництві.

Зміст

[Створення або імпорт аплету](#)

[Запуск і налагодження аплету](#)

[Вбудовування аплету у веб-додаток](#)

Створення або імпорт вихідного файлу аплета

У цьому розділі описана процедура створення першого аплета. Можливо, вже є готовий аplet; в цьому випадку можна імпортувати його в середу IDE за допомогою послідовності дій, наведеної нижче.

Створення проекту Java "з нуля" або з існуючих вихідних кодів

Виберіть "File> New Project" (Ctrl-Shift-N). В області "Categories" виберіть "General".

Виберіть один з наступних варіантів:

Якщо створюється новий вихідний файл аплета, виберіть "Java Class Library" в "Projects". Натисніть кнопку "Next".

Якщо потрібно імпорт вихідного файлу аплета, виберіть "Java Project with Existing Sources". Натисніть кнопку "Next". Вкажіть розташування файлу в текстовому полі "Source Packages Folder".

У полі "Project Name" введіть HelloApplet. У полі "Project Location" вкажіть будь-яку папку на комп'ютері.

Натисніть кнопку "Finish". Якщо вихідний файл аплета був імпортований, [запустіть його і виконайте налагодження](#)

Створення вихідного файлу апплета

- Клацніть правою кнопкою миші вузол проекту "HelloApplet" у вікні "Projects" і виберіть "New File / Folder" (Ctrl-N).
- В області "Categories" виберіть "Java Classes". В області "File Types" виберіть "Applet". Як альтернатива, якщо передбачається візуальне проектування апплету, виберіть "Java GUI Forms> AWT Forms> Applet Form".
- Натисніть кнопку "Next".
- У полі "Under Class Name" введіть MyApplet. У полі "Package" уведіть org.me.hello.
- Натисніть кнопку "Finish".
У середовищі IDE автоматично створюється вихідний файл апплета в зазначеному пакеті. Вихідний файл апплета відкривається в редакторі вихідного коду.

-

- Визначте клас аплету шляхом копіювання та вставки наступного коду замість існуючого коду за замовчунням:

```
package org.me.hello;
```

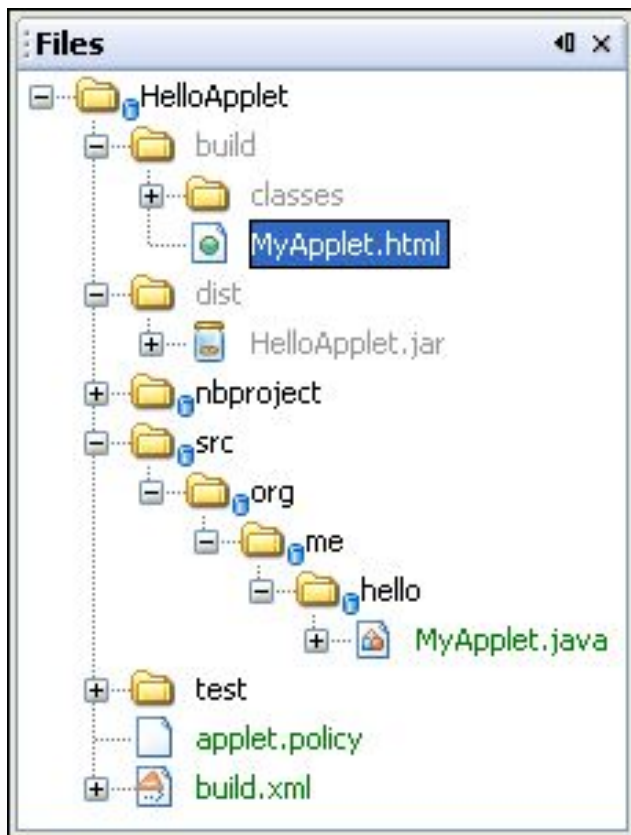
```
import java.applet.Applet;  
import java.awt.Graphics;
```

```
public class MyApplet extends Applet {  
    public void paint (Graphics g) {  
        g.drawString ("Hello applet!", 50, 25);  
    }  
}
```

Запуск і налагодження вихідного файлу аплета

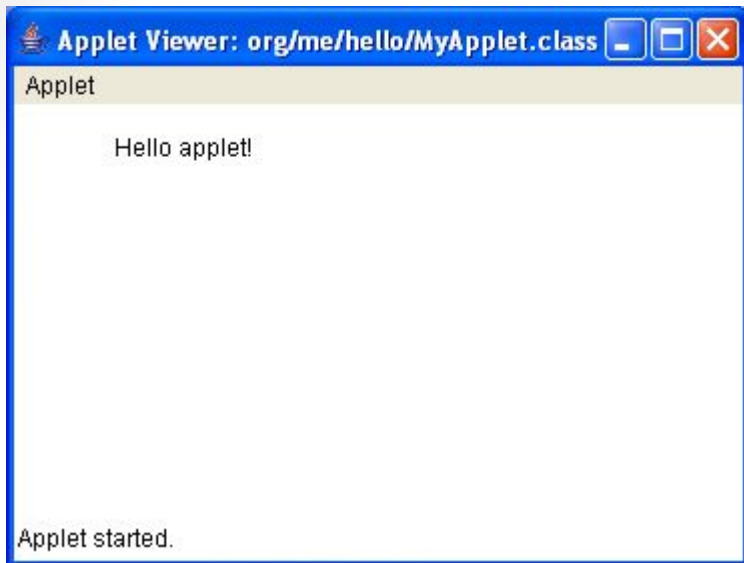
- Запуск і налагодження створеного аплета можна виконувати в середовищі IDE. У цьому розділі наводиться відповідна процедура.
- Збирання і запуск вихідного файлу аплета
- Клацніть правою кнопкою миші вузол проекту у вікні "Projects" і виберіть "Build Project" з контекстного меню.
- У dist папці створюється файл HelloApplet.jar.
- Клацніть правою кнопкою миші вузол класу аплету у вікні "Projects" і виберіть "Run File" з контекстного меню.
-

- У папці build створюється файл кошти запуску MyApplet.html з вбудованим аплетом:

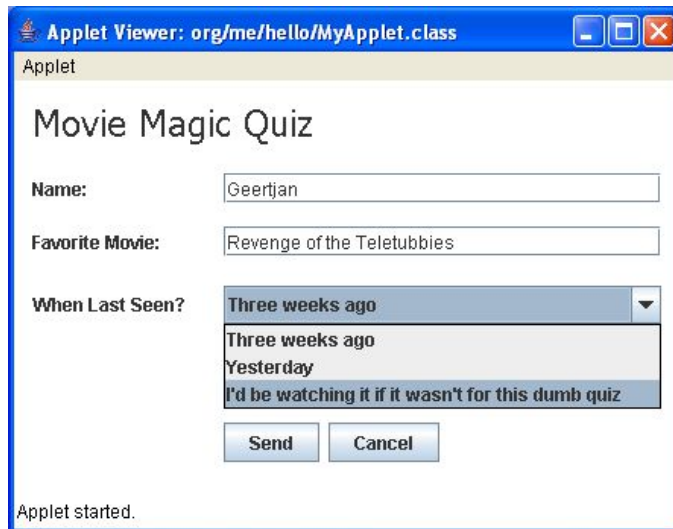


Перегляд аплета

Аплет запускається в програмі перегляду "Applet Viewer":



У програмі перегляду "Applet Viewer" також відображаються форми аплета:



Налагодження вихідного файлу аплету за допомогою зміни його параметрів

- Файл кошти запуску MyApplet.html в папці build перезаписується при кожному запуску або налагодженні аплету. Тому файл кошти запуску в папці build не слід змінювати. Замість цього використовується наступна процедура:
- Відкрийте вікно "Files" (Ctrl-2).
- Скопіюйте файл кошти запуску MyApplet.html з папки build в пакет, де розташовується клас аплету, в папці src (в даному випадку - в org.me.hello). Переконайтеся в тому, що ім'я файлу кошти запуску MyApplet.html збігається з ім'ям класу аплету.
- Потім змініть файл кошти запуску MyApplet.html відповідно до наявних вимог. При складанні проекту файл кошти запуску MyApplet.html копіюється з папки в папку src build.
- **Рекомендація:** Файл кошти запуску створюється в середовищі IDE при запуску налагодженні аплету. Якщо його скопіювати для зміни в папку src, він автоматично додається в файл JAR при складанні проекту. Як правило, додавати цей файл при упаковці програми не потрібно. Для виключення файлу кошти запуску з файлу JAR необхідно натиснути правою кнопкою миші проект, вибрати "Properties", клацнути "Packaging" і додати вираз для виключення файлів кошти запуску, таких як MyApplet.html.

Вбудовування аплету у веб-додаток

Аплет готовий. Тепер необхідно забезпечити його доступність для користувача. Для цього створюється веб-додаток, JAR аплету міститься в дорогу до класів, і в файл HTML веб-додатки додається тег аплету.

Створення веб-проекту

Виберіть "File> New Project". В області "Categories" виберіть "Web". В області "Projects" виберіть "Web Application". Натисніть кнопку "Next".

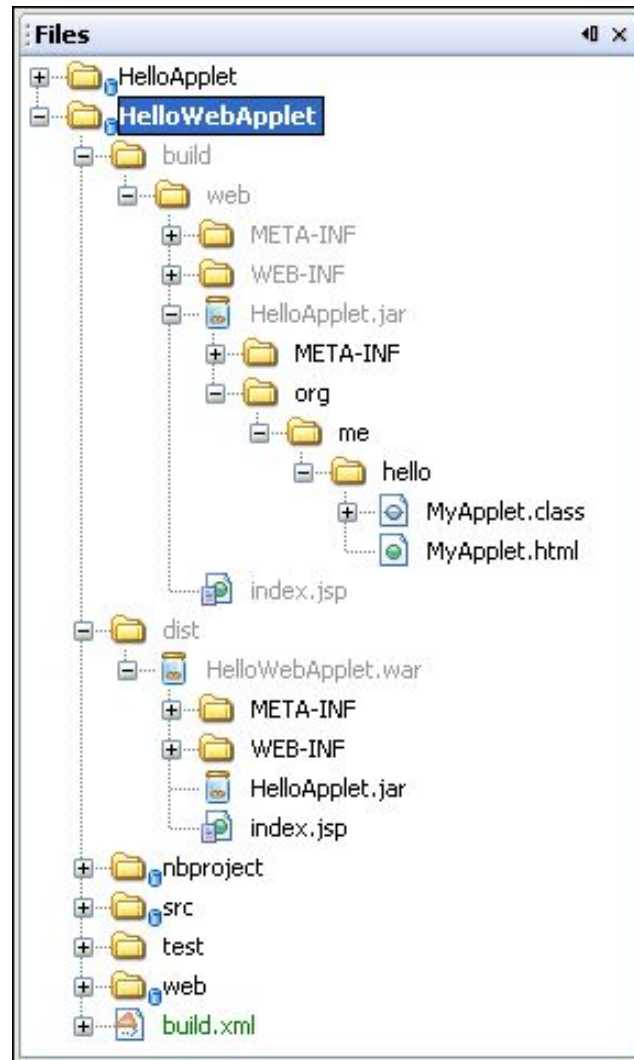
У полі "Project Name" введіть HelloWebApplet. У полі "Project Location" вкажіть будь-яку папку на комп'ютері. Натисніть кнопку "Finish".

Додавання файлу JAR аплету до веб-проекту

Якщо потрібно включити файл JAR аплету у веб-проект, цього можна добитися шляхом додавання проекту Java, що містить файл JAR, або шляхом додавання самого файлу JAR. Вибір залишається за розробником, проте слід враховувати, що при додаванні проекту Java до веб-проекту стає можливою автоматичне складання аплету при складанні веб-додатки. Таким чином, при зміні аплету в проекті Java при кожній збірці веб-проекту здійснюється складання і нової версії аплету. З іншого боку, якщо файл JAR аплету не перебуває в проекті Java середовища IDE NetBeans 5.x, вихідний файл аплету при складанні веб-проекту не піддається повторної збірці.

1. У вікні "Projects" клацніть правою кнопкою миші вузол проекту "HelloWebApplet" і виберіть пункт "Properties" в контекстному меню.
2. Виберіть один з наступних варіантів:
 - a) Якщо файл JAR аплета знаходиться в проекті Java, клацніть "Packaging", а потім "Add Project". Перейдіть до папки, яка містить проект Java, та виберіть її. Зверніть увагу, що проекти середовища IDE відзначені значком проекту середовища IDE NetBeans.
 - b) Якщо файл JAR аплета не перебуває в проекті IDE, клацніть "Packaging", а потім "Add JAR / Folder". Перейдіть до папки, яка містить файл JAR, та виберіть її.
3. Файл JAR, що містить вихідний файл аплета, наводиться в таблиці в нижній частині діалогового вікна "Project Properties".
4. Також можна вказати розташування аплету у стовпці "WAR" таблиці "Path". За замовчуванням файл JAR аплета копіюється в корінь документа веб-додатки, тобто в папку build / web (вищий рівень структури файлів веб-додатки). Натисніть кнопку "OK".

При складанні проекту файл JAR аплета упаковується в файл WAR проекту в папці dist. Він також додається в папку build / web. Для отримання більш докладної інформації див. ілюстрацію нижче.



Створення та виконання файлу JSP або HTML

Виберіть один з наступних варіантів:

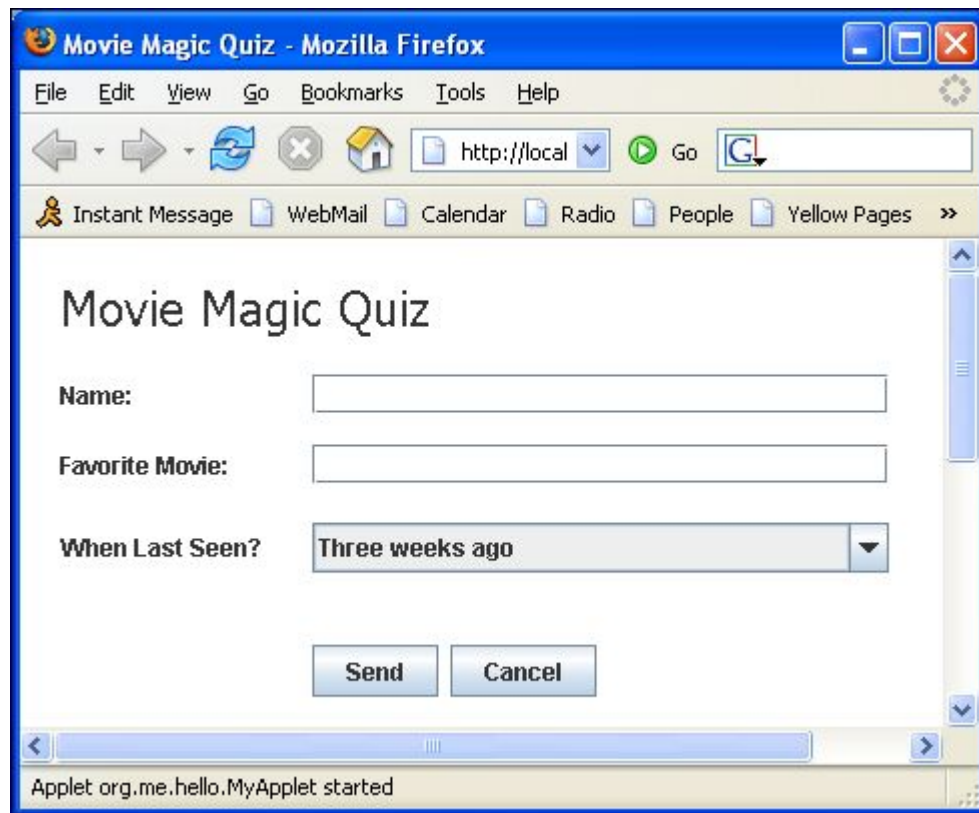
- Якщо аплет потрібно вмонтувати в файл JSP, двічі клацніть файл за замовчуванням index.jsp у вікні "Projects". Цей файл створюється в середовищі IDE при створенні веб-проекту. Файл відкривається в редакторі вихідного коду.
- Якщо аплет потрібно вмонтувати в файл HTML, клацніть правою кнопкою миші вузол проекту "HelloWebApplet" і виберіть "New> File / Folder" з контекстного меню. В області "Categories" виберіть "Web". В області "File Types" виберіть "HTML". Натисніть кнопку "Next". Дайте файлу HTML ім'я та натисніть кнопку "Finish".
- Вбудуйте аплет в файл шляхом додавання наступного тега аплету в будь-якому місці між тегами <body> </ body> у файлі:

```
<applet code="org.me.hello.MyApplet" archive="HelloApplet.jar"/>
```

- Повне ім'я класу аплету - org.me.hello.MyApplet.
- Аплет міститься у файлі JAR HelloApplet.jar.

- Клацніть правою кнопкою миші вузол JSP або HTML у вікні "Projects" і виберіть "Run File" з контекстного меню.
Файл JSP або HTML розгортається з сервера у браузері середовища IDE за замовчуванням.

Результат повинен бути подібний до наведеного нижче:



У разі форм аплету результат схожий на наступний малюнок (успішне тестування проведено тільки в FireFox):

СЕРВЛЕТИ, СТОРІНКИ JAVASERVER PAGES, WEB СЕРВЕР ТОМКАТ

Короткі відомості про сервлети і сторінки JavaServer Pages

- *Що таке сервлети*
- *Коли й чому варто використати сервлети*
- *Що таке сторінка JavaServer Pages (JSP)*
- *Коли й чому використовують JSP*
- *Як і де одержати програмне забезпечення сервлетів й JSP*
- *Установка й настроювання програмного забезпечення*

Лекція являє собою короткий огляд сервлетів і сторінок JavaServer Pages (JSP) з оглядом їхніх основних переваг. У ній також приводяться відомості про одержання й установку програмного забезпечення, що потрібно для написання сервлетів і розробки документів JSP.

Сервлети

Появою сервлетів Java-технології відреагували на програмування Common Gateway Interface (CGI). Сервлети - це програми, які виконуються на Web-сервері, діючи як посередник між запитом, що надходить від Web-браузера або іншого клієнта HTTP, і базами даних або додатками на HTTP-сервері. Вони виконують наступні дії:

1. Читають будь-які дані, передані користувачем.

Дані звичайно вводяться у форму на Web-сторінці, але також можуть надійти від аплету Java або набудовує HTTP- програми, що, клієнта.

2. Переглядають всю інформацію про запит, що убудована в HTTP-запит.

Інформація включає відомості про можливості браузера, cookies, імені хоста клієнта, що робить запит, і т.д.

3. Генерують результати.

Цей процес може зажадати спілкування з базою даних, виконання виклику RMI або CORBA, активізації існуючого додатка або безпосереднього обчислення відповіді.

4. Форматують результати усередині документа.

У більшості випадків цей процес припускає вбудовування інформації на HTML-сторінку.

5. Установлюють відповідні параметри HTTP-відповіді.

Це означає, що браузеру надходить повідомлення про тип документа, що повертає, (наприклад, HTML), установці cookies, кешированих параметрів й інших подібних дій.

6. Повертають документ клієнтові.

Документ може бути повернутий клієнтові в текстовому форматі (HTML),

Переваги сервлетів у порівнянні з «традиційними» CGI

Сервлети Java ефективні, прості у використанні; вони є могутнішими, стерпними, безпечними й до того ж дешевше, ніж традиційні CGI і багато альтернативних CGI-подібних технологій.

Ефективність

При використанні традиційного CGI для кожного HTTP-запиту запускається свій процес. Якщо CGI-програма сама по собі є короткої, то накладні витрати по запуску процесу непорівнянно великі в порівнянні згодом її виконання. При використанні сервлетів Java Virtual Machine (віртуальна Java-машина) залишається в режимі виконання й обробляє кожен запит, використовуючи спрощений Java-потік, а не складний процес операційної системи. До того ж якщо в традиційному CGI до однієї й тій же програмі одночасно надходить N запитів, код CGI-програми завантажується на згадку N раз. При використанні сервлетів одночасно може існувати N потоків, але в пам'яті буде перебувати єдина копія класу сервлета. І, нарешті, коли CGI-програма закінчує обробку запиту, програма завершує виконання. Це ускладнює кешування обчислень, підтримка відкритими з'єднань із базою даних і виконання інших оптимізаційних дій, які засновані на сталості існування даних. Сервлети же залишаються в пам'яті навіть після виконання запиту, що дає можливість довільним образом зберігати складні дані між запитами.

Зручність

Сервлети мають розширену інфраструктуру для здійснення автоматичного синтаксичного аналізу й декодування даних HTML-форм, читання й завдання HTTP-заголовків, обробки cookies, відстеження сеансів і багатьох інших подібних високий рівень утиліт. Крім того, ви вже знаєте мову програмування Java. Навіщо ще вчити Perl? Ви вже переконалися, що Java-технологія створює більше надійний повторно

Потужність

Сервлети мають деякі можливості, які складно або неможливо представити у звичайному CGI-програмуванні. Вони можуть спілкуватися безпосередньо з Web-сервером, у той час як звичайні CGI-програми не можуть цього робити, принаймні без використання особливого, залежного від сервера, API. А можливість спілкування із сервером спрощує, наприклад, перетворення відносних адрес URL у правильні, повні імена шляхів. Крім того, трохи сервлетів можуть спільно використати дані, спрощуючи реалізацію організації пула з'єднань із базою даних й інші методи оптимізації спільного використання ресурсів. Також сервлети можуть зберігати інформацію між запитами, спрощуючи реалізацію таких технологій, як відстеження сеансів і кэширування попередніх обчислень.

Переносимість

Сервлети пишуться мовою програмування Java і мають стандартний інтерфейс. Отже, сервлети, написані, скажемо, для сервера I-Planet Enterprise Server, можуть запускатися на серверах Apache, Microsoft Internet Information Server (IIS), IBM WebSphere або StarNine WebStar без внесення яких-небудь змін. Так, всі сервлети й сторінки JSP, представлені в книзі, були виконані на серверах Java Web Server компанії Sun, Apache Tomcat й JavaServer Web Development KIT (JSWDK) без єдиної зміни коду. Багато хто з них також були протестовані на серверах WebLogic компанії BEA й WebSphere компанії IBM. Дійсно, сервлети підтримуються безпосередньо або через програми, що підключають додатково (plug-ins) на всіх основних Web-серверах. У цей час вони є частиною платформи Java 2 Platform, Enterprise Edition (промислове видання) (J2EE, див. <http://java.sun.com/j2ee/>). Тобто промислова підтримка

Безпека

Однієї з головних причин уразливості традиційних CGI-програм є те, що вони часто виконуються в оболонках операційних систем загального призначення. Тому програміст повинен бути дуже уважним й отфільтровувати символи, які особливим образом трактуються оболонкою, наприклад, зворотні лапки й крапки з коми. Це набагато складніше, ніж здається з першого погляду, і слабкі сторони програм, що виникають із цієї причини, постійно виявляються в широко використовуваних бібліотеках CGI. Другим джерелом проблем є те, що деякі CGI-програми обробляються мовами, які не перевіряють автоматично границі масивів або рядків. Наприклад, у мовах програмування C и C++ можна зовсім законно розмістити в пам'яті масив з 100 елементів, а потім записати значення в 999-й «елемент», що насправді є деякою випадковою областю пам'яті програми. Тому, якщо програмісти забувають перевіряти вихід за межі масивів, вони відкривають системи для навмисних або випадкових атак переповнення буфера. У сервлетів не існує подібних проблем. Навіть якщо для активізації програми, що перебуває в локальній операційній системі, сервлет виконує вилучений системний виклик, він не використовує для цього оболонку. І звичайно, перевірка границь масиву, як й інші функції захисту пам'яті, є найважливішою частиною мови програмування Java.

Вартість

У цей час є безкоштовні або дуже недорогі Web-сервери, зручні для «персонального» використання або створення Web-сайтів невеликого обсягу. Однак за винятком сервера Apache, що є безкоштовним, більшість Web-серверів комерційного рівня якості є відносно дорогими. Проте, якщо у вас є Web-сервер, те незалежно від його ціни додавання підтримки сервлетів не обійдеться вам дорого. Це відрізняє сервлети від багатьох альтернатив CGI, які вимагають значних початкових інвестицій на придбання

Сторінки JavaServer Pages

Технологія JavaServer Pages (JSP) дозволяє змішувати звичайний, статичний HTML з динамічно генерованим змістом сервлетів. Багато Web-сторінок, які побудовані CGI-програмами, споконвічно є статичними, що мають невеликі частини, що змінюються в обмежених межах. Наприклад, вихідна сторінка більшості онлайн-магазинів однакова для всіх відвідувачів, за винятком невеликого вітання, що включає ім'я відвідувача, якщо воно відомо. Але більшість варіантів CGI-програм, включаючи сервлети, змушують вас генерувати в програмі всю сторінку, навіть якщо її більша частина залишається незмінною. Сторінки JSP дозволяють створювати обидві частини роздільно. У лістингу 1.1 наведений приклад. Більша частина сторінки складається зі звичайного HTML-коду, що передається відвідувачеві без змін. Частини, які генеруються динамічно, відзначені спеціальними HTML-подібними тегами й перебувають прямо на сторінці.

Приклад JSP-сторінки

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Welcome to Our Store</TITLE></HEAD> .
<BODY>
<H1>We1come to Our Store</H1>
<SMALL>Welcome.
<!--Ім'я користувача при першому відвідуванні "New User" -і>
<*= Utils.getUserNameFromCookie(request) *>
Щоб одержати доступ до параметрів облікового запису, клацніть
<A HREF="Account-Setti ngs. html ">тут.</A></SMALL>
<P>
Звичайний HTML для всієї іншої частини Web-сторінки
онлайнового магазину
</BODY>
</HTML>
```

Переваги JSP

Сторінки JSP мають ряд переваг у порівнянні з іншими альтернативами CGI. Розглянемо деякі з них.

Порівняння з Active Server Pages (ASP)

Технологія ASP компанії Microsoft є конкурентом сторінок JSP. Але в JSP удвічі більше переваг. По-перше, динамічна частина пишеться на Java, а не на VBScript або іншому, характерному для ASP мові, тому є могутнішою й більше підходящою для складних додатків, які вимагають повторно використовуваних компонентів. По-друге, JSP переносимо в інші операційні системи й Web-сервери, не обмежуючись Windows NT/2000 й IIS. Ті ж аргументи можна привести при порівнянні JSP з ColdFusion. Сторінки JSP дозволяють використати Java і не прив'язані до конкретного сервера.

Порівняння із PHP

Мова PHP є безкоштовним, відкритим, що вбудовують в HTML мовою програмування, що деякою мірою аналогічний й ASP, і JSP. Перевага JSP полягає в тому, що динамічна частина написана мовою Java, що ви, можливо, уже знаєте, і який має розширений API для роботи в мережі, доступу до баз даних, розподіленим об'єктам і т.п., у той час як використання PHP вимагає вивчення зовсім нової мови.

Порівняння зі звичайними сервлетами

Сторінки JSP не надають таких можливостей, які в принципі не могли б бути виконані сервлетами. Дійсно, документи JSP автоматично переводяться в сервлети. Однак зручніше писати (і модифікувати!) звичайний HTML-код, чим мати справу з безліччю операторів `print "In,` які генерують HTML. Крім того, поділ умісту й подання дозволяє розділяти завдання між різними людьми. Експерти по дизайні Web-сторінок можуть створювати HTML, використовуючи звичний інструментарій і залишаючи місце для програмістів сервлетів, які вставляють згодом на < сторінку динамічний уміст.

Одержання програмного забезпечення сервлетів й JSP

Ваш перший крок - це завантаження програмного забезпечення, що реалізує Java Met 2.1 або 2.2 й JavaServlet Pages 1.0 або 1.1. Якщо ви використаєте сучасний сервер, то існує ймовірність, що в ньому є все, що вам потрібно. Перевірте документацію сервера або перегляньте останній список серверів, які підтримують сервлети на сайті <http://java.sun.com/products/servlet/industry.html>. Хоча ви, безперечно, захочете працювати із сервером комерційного рівня якості, починати вивчення корисно, маючи безкоштовну систему, яку можна встановити на настільному комп'ютері для розробки й тестування сервлетів. Приведемо найбільш відомі приклади таких систем.

• Apache Tomcat.

Tomcat є офіційною довідковою реалізацією специфікацій Servlet 2.2 й JSP 1.1. Його можна використати як невеликий автономний сервер для тестування сервлетів і сторінок JSP або інтегрувати його в Apache Web-сервер. Але й інші сервери оголосили про майбутню підтримку цих специфікацій, тому вони докладно розглядаються в книзі. Сервер Tomcat, як й Apache, є безкоштовним. Однак, як й Apache (який є дуже швидким, високо надійним, але складним в установці й конфігуруванні сервером), Tomcat вимагає значно більших зусиль при установці, чим процесори сервлетов у комерційних Web-серверах. Подобиці наведені на сайті <http://jakarta.apache.org/>.

• JavaServer Web Development Kit (JSWDK).

JSWDK є офіційною довідковою реалізацією специфікацій сервлетів 2.1 й JSP 1.0. Використається як невеликий автономний сервер для тестування сервлетів і сторінок JSP перед їхнім розміщенням на повномасштабному Web-сервері, що підтримує ці технології. JSWDK — це безкоштовний і надійний продукт, але його установка й конфігурування вимагають деяких зусиль. Більш докладно питання розгляньте на <http://java.sun.com/products/servlet/download.html>.

- **Allaire JRun.**

JRun — це ядро (програма) підтримки сервлетів й JSP, що може бути убудоване в сервер Netscape Enterprise або FastTrack, IIS, Microsoft Personal Web Server, більше старі версії Apache, WebSite компанії O'Reilly або StarNine WebStar. Скорочена версія, що підтримує до п'яти одночасних з'єднань, є безкоштовною; у комерційній версії це обмеження відсутнє, вона має додаткові можливості, наприклад, вилучену консоль адміністрування. Подобиці можна знайти на сайті <http://www.allaire.com/products/jrun/>.

- **LiteWebServer (LWS) компанії Gefion Software.**

LWS — це невеликий безкоштовний Web-сервер на базі Tomcat, що підтримує сервлети версій 2.2 й JSP 1.1. Gefion також має безкоштовну підключати програму, Що, WAICoolRunner, що забезпечує підтримку сервлетів 2.2 й JSP 1.1 серверами Netscape FastTrack й Enterprise. Див. сайт <http://gefionsoftware.com/>.

- **Java Web Server компанії Sun.**

Цей сервер повністю написаний мовою Java і був одним з перших Web-серверів з повною підтримкою специфікацій сервлетів 2.1 й JSP 1.0. Хоча більше він активно не розробляється, тому що компанія Sun зосереджує увагу на сервері Netscape/1-Planet, він усе ще популярний при вивченні сервлетів й JSP. Інформація про безкоштовну пробну версію наведена за адресою <http://www.sun.com/software/jwebserver/try/>. Відомості про безкоштовну версію без обмеження строку роботи для цілей навчання в академічних інститутах перебувають на <http://freeware.thesphere.com/>.







